

The rectilinear Steiner tree problem with given topology and length restrictions

Jens Maßberg

Institut für Optimierung und Operations Research, Universität Ulm,
jens.massberg@uni-ulm.de

Keywords: Steiner trees with given topology, rectilinear Steiner trees, dynamic programming, totally unimodular, shallow light Steiner trees

Abstract. We consider the problem of embedding the Steiner points of a Steiner tree with given topology into the rectilinear plane. Thereby, the length of the path between a distinguished terminal and each other terminal must not exceed given length restrictions. We want to minimize the total length of the tree.

The problem can be formulated as a linear program and therefore it is solvable in polynomial time. In this paper we analyze the structure of feasible embeddings and give a combinatorial polynomial time algorithm for the problem. Our algorithm combines a dynamic programming approach and binary search and relies on the total unimodularity of a matrix appearing in a sub-problem.

1 Introduction

The RECTILINEAR STEINER TREE PROBLEM WITH GIVEN TOPOLOGY AND LENGTH RESTRICTIONS can be stated as follows. The input (S, T, r, p, l) consists of a set of terminals T with positions $p : T \rightarrow \mathbb{R}^2$, a tree S with $T \subseteq V(S)$, a distinguished terminal $r \in T$ - called the *root* of the tree - and length restrictions $l_t \in \mathbb{R}_{\geq 0}$ for all $t \in T$.

The task is to find an embedding $\pi : V(S) \rightarrow \mathbb{R}^2$ of the vertices of the tree into the plane with $\pi(t) = p(t)$ for all $t \in T$, such that for all $t \in T$ the length $d_\pi(t)$ of the unique path from r to t in S with edge set $E_S[r, t]$ has length at most l_t , that is,

$$d_\pi(t) = \sum_{\{v,w\} \in E_S[r,t]} \|\pi(v) - \pi(w)\|_1 \leq l_t \quad (1)$$

and the total length

$$c(\pi) := \sum_{\{v,w\} \in E(S)} \|\pi(v) - \pi(w)\|_1 \quad (2)$$

of the tree is minimized. The tree S is called *Steiner tree* and the vertices in $V(S) \setminus T$ *Steiner points*. Throughout this paper we assume w.l.o.g. that the

root is placed at the origin, that is, $p(r) = (0, 0)$. By adding Steiner points and edges of length zero we can assume that the terminals are leaves of S and that all Steiner points have degree 3. Moreover, we denote by $\pi_x(v)$ and $\pi_y(v)$ the x - and y -coordinate, respectively, of $\pi(v)$ for an embedding π and a vertex $v \in V(S)$.

A further generalization of the problem is to extend it to other metrics or to consider length restrictions between any pair of terminals. In this paper we restrict ourselves to the ℓ_1 metric and length restrictions between one distinguished vertex and all other terminals, as this case has a strong application in practice.

Our problem is motivated by an application arising in VLSI design, where one of the main challenges is to build so-called repeater trees. These are tree-like structures consisting of wires and possibly so-called repeater circuits and their task is to distribute a signal from a source circuit to several sink circuits. Thereby, the signal is delayed. In order to guarantee, that the chip works on the desired speed, timing constraints are given, that is, the signal has to arrive at each sink circuit not later than a given individual time bound. There are several heuristics to build such repeater trees (see e.g. [1]).

A repeater tree can be modeled as a Steiner tree connecting the source and the sinks and containing repeater circuits at some of the Steiner points. The length of a repeater tree corresponds to its power consumption. So the question arises, if the length of a given tree can be reduced by moving the positions of the Steiner points. Bartoschek et al. [1] have shown, that by adding repeater circuits at appropriate positions the delay of a signal on a path from the source to a sink is approximately proportional to the length of the path. Thus the timing constraints directly yield length restrictions on root-terminal paths. It turns out, that the *Rectilinear Steiner Tree Problem with given Topology and Length restrictions* is a good model for the task to minimize the power consumption of given repeater trees without changing their topology.

If we are allowed to change the topology of the tree, the problem becomes NP-hard, as it contains the *Rectilinear Steiner Tree Problem* [5]. If, additionally, $l_t = \|p(s) - p(t)\|_1$ for all $t \in T$, that is, all root-terminal paths are shortest paths, we end at the *Rectilinear Steiner Arborescence Problem*, which is also NP-hard ([12,11]). In the case where the length restrictions are the same for all terminals we have the case of *Shallow Light Steiner Trees*.

However, if we have to keep the topology, but do not have any length restrictions, an optimal embedding can be computed in linear time using dynamic programming (see e.g. [9]). To our knowledge, the problem of embedding a Steiner tree with a given topology satisfying length restrictions has not been considered yet. In this paper we present the first combinatorial polynomial time algorithm that computes an optimal embedding.

Figure 1 (i) shows an instance with seven terminals drawn as black squares and 5 Steiner points drawn as white circles. Figure (ii) shows an optimal solution if there are no length restrictions. In Figure (iii) an optimal solution is shown, if we have length restrictions $l_{t_1} = 5$, $l_{t_2} = 6$ and $l_s = \infty$ otherwise. If there are no length restrictions, then there always exists an optimal solution where

the Steiner points are positioned at the so called Hanan grid on T (see [7]). With length restrictions, this is no longer true. Nevertheless, we prove that if the positions of the terminals and the length restrictions are integral, then there always exists an solution on half-integral positions.

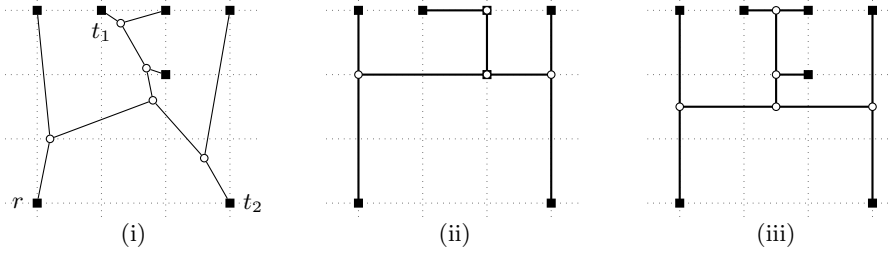


Fig. 1. Instance (i), optimal embedding without length restrictions (ii) and optimal embedding with length restrictions $l_{t_1} = 5$ and $l_{t_2} = 6$ (iii). The regular dotted grid has a lattice spacing of 1.

The problem can be formulated as a linear program by extending the LPs presented in [3,9]. Therefore it can be solved in polynomial time by non-combinatorial algorithms. Nevertheless, we are interested in a combinatorial algorithm for the problem.

After introducing several definitions concerning the movement of components of the tree in Section 2, we present our main observations in Section 3. Among others, we prove that there always exists an optimal embedding where the Steiner points are on half-integral positions. Based on this observation, we introduce in Section 4 a dynamic programming algorithm which is the main ingredient to achieve a pseudo-polynomial time algorithm. Refining this algorithm we finally gain a polynomial time algorithm in Section 5.

2 Moving Components

Before we come to the main observations of the paper we examine how the movements of Steiner points of a given embedding influence the total length of the tree and the length of root-terminal paths. First we start with several definitions that we need throughout this paper.

If π is an embedding, then an x -component C at position $x(C)$ with respect to π , $x(C) \in \mathbb{R}$, is a connected subtree C of T such that all vertices in C have x -coordinate $x(C)$. An x -component C is called *maximal* if there does not exist any x -component C' with $C \subsetneq C'$. A component always depends on the embedding π . In the following, we omit π in the notation if it is clear from the context. In an analogous way we define a y -component C at position $y(C)$. In the remainder of the paper we introduce several definitions and state lemmata

concerning x -components. By symmetry, these definitions and lemmata also hold for y -components.

Let $\Gamma(V(C))$ be the neighbors of the vertices of C . For an x -component C we define

$$\Gamma_{<}^\pi(C) := \{v \in \Gamma(V(C)) : \pi_x(v) < x(C)\} \text{ and} \quad (3)$$

$$\Gamma_{>}^\pi(C) := \{v \in \Gamma(V(C)) : \pi_x(v) > x(C)\}. \quad (4)$$

In an analogous way we define $\Gamma_{<}^\pi(C)$ and $\Gamma_{>}^\pi(C)$ for a y -components C . If C is a component not containing r , then the *predecessor* of C is the unique vertex $v \in \Gamma_{>}^\pi(C) \cup \Gamma_{<}^\pi(C)$ such that v is on the root- w path for all $w \in V(C)$. For simplicity of notation we define

$$\text{sign}(C) = \begin{cases} 1 & \text{if the predecessor of } C \text{ is in } \Gamma_{<}^\pi(C) \\ -1 & \text{otherwise.} \end{cases} \quad (5)$$

If C is an x -component with respect to some embedding π then we say that we *move C by δ* if we replace π by the embedding π' defined by

$$\pi'(v) := \begin{cases} \pi(v) + (0, \delta) & \text{for all } v \in V(C) \setminus T, \\ \pi(v) & \text{otherwise.} \end{cases} \quad (6)$$

We say, that we move C *towards its predecessor* if $\delta \cdot \text{sign}(C) < 0$.

If C is a maximal component containing no terminals, then we define $R(C)$ to be the set of terminals t such that the unique root- t path P passes C , that is, $V(P) \cap V(C) \neq \emptyset$ and the path enters and leaves C at the same side, that is, we have either $|V(P) \cap \Gamma_{>}^\pi(C)| = 2$ or $|V(P) \cap \Gamma_{<}^\pi(C)| = 2$. If we choose $\delta \in \mathbb{R}$ with $|\delta|$ small enough and move C by δ , then the length of all root- t paths with $t \in R(C)$ change by $2\text{sign}(C)\delta$. The length of any other root-terminal path does not change.

Figure 2 illustrates some of the definitions.

If π and π' are two embedding, then we say that π' *preserves the local order of π* if for every edge $(v, w) \in E(S)$ we have

$$(\pi_x(v) \leq \pi_x(w)) \Rightarrow (\pi'_x(v) \leq \pi'_x(w)) \text{ and} \quad (7)$$

$$(\pi_y(v) \leq \pi_y(w)) \Rightarrow (\pi'_y(v) \leq \pi'_y(w)). \quad (8)$$

Note, that by (7) if $\pi_x(v) = \pi_x(w)$ then $\pi'_x(v) = \pi'_x(w)$ and analogously for y . This implies that each component with respect to π is also a component with respect to π' (but not necessarily the other way round!). Moreover, if v is a vertex of an x -component that contain terminals, we have $\pi_x(v) = \pi'_x(v)$.

Now we can analyze how the length of the embedding and of root-terminal paths change if we move maximal components simultaneously and the local order is preserved.

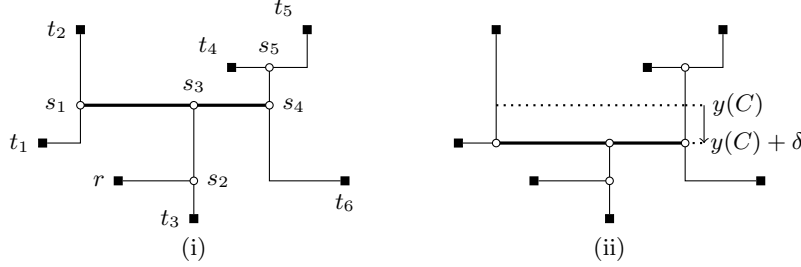


Fig. 2. (i) An embedding π with a maximal y -component C with $V(C) = \{s_1, s_3, s_4\}$, predecessor s_2 , $\Gamma_>(C) = \{t_2, s_5\}$, $\Gamma_<(C) = \{t_1, s_2, t_6\}$, $\text{sign}(C) = 1$ and $R(C) = \{t_1, t_6\}$. (ii) Embedding obtained by moving C by $\delta < 0$. The new embedding preserves the local positions of π . The length of all root- t with $t \in R(C)$ changed by $2\text{sign}(C)\delta = 2\delta$.

Lemma 1. Let π be an embedding, Δ be the set of all maximal x - and y -components, and $\delta_C \in \mathbb{R}$ for $C \in \Delta$. Denote by π' the embedding we obtain by moving each component $C \in \Delta$ by δ_C . If π' preserves the local order of π then

$$c(\pi') = c(\pi) + \sum_{C \in \Delta} \delta_C (|\Gamma_<^\pi(C)| - |\Gamma_>^\pi(C)|). \quad (9)$$

Moreover we have for all $t \in T$:

$$d_{\pi'}(t) = d_\pi(t) + \sum_{C \in \Delta: t \in R(C)} 2\text{sign}(C) \cdot \delta_C. \quad (10)$$

Proof. Consider an x -component $C \in \Delta_x$. If we move C , then only the length of edges $\{v, w\} \in E(S)$ with $v \in V(C)$ and $w \notin V(C)$ are changed. Let $\{v, w\}$ be such an edge and assume $w \in \Gamma_<^\pi(C)$, that is $\pi_x(w) < \pi_x(v)$. As the local order is preserved, we have $\pi'_x(w) \leq \pi'_x(v)$. But then moving C by δ increases the length of the edge $\{v, w\}$ by δ . In an analogous way we see that the length of the edge decreases by δ if $w \in \Gamma_>^\pi(C)$. Summing up the changes over all components we obtain (9).

Now consider a terminal $t \in T$. Again, as the local order is preserved by π' , the length of the root- t path is only influenced by components C with $t \in R(C)$. Consider such a component C . If we move C by $|\delta_C|$ towards the predecessor of C , the length of the path is reduced by $2|\delta_C|$. On the other hand, if we move C in the other direction by $|\delta_C|$, then the length is increased by $2|\delta_C|$. In total, the length changes by $\text{sign}(C)2 \cdot \delta_C$. Summing up over all such components, we obtain (10). \square

The following observation is crucial in order to prove that there exist optimal solutions that are half-integral.

Lemma 2. If Δ is a set of maximal x -components that do not contain terminals, then $\{R(C)\}_{C \in \Delta}$ is a laminar family.

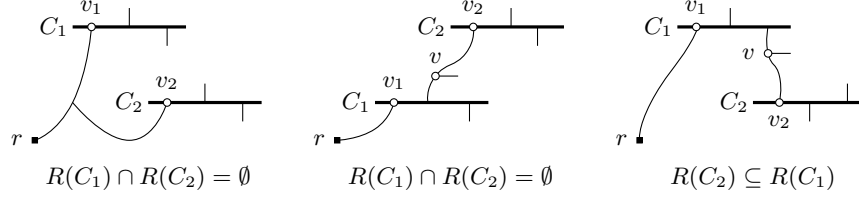


Fig. 3. The possible positions of two maximal y -components within a tree.

Proof. Let $C_1, C_2 \in \Delta$. By definition $V(C_1) \cap V(C_2) = \emptyset$. For $i \in \{1, 2\}$ let v_i be the vertex of $V(C_i)$ that is adjacent to the predecessor of C_i . Note that v_i is on the unique r - t path for every $t \in R(C_i)$ (see Figure 3). Now assume that neither v_1 is on the r - v_2 path nor v_2 is on the r - v_1 path. Then $R(C_1) \cap R(C_2) = \emptyset$. Otherwise, assume w.l.o.g. that v_1 is on the r - v_2 path. In this case there exists a unique vertex v on the v_1 - v_2 path satisfying $v \in F_{>}^\pi(C_1) \cup F_{<}^\pi(C_1)$. Now note that the length of all r - t paths change for all $t \in R(C_2)$ when moving C_1 if and only if the length of the root- t path changes when moving C_1 . Hence, $R(C_2) \subseteq R(C_1)$ or $R(C_2) \cap R(C_1) = \emptyset$. This implies the desired result. \square

Before we continue with the main result we make another simple observation:

Proposition 1. *If π is an embedding and there exists a vertex $t \in T$ such that $d_\pi(t) > \|p(v) - p(r)\|_1$, then there exists a component C such that moving C towards its predecessor decreases the length of the root- t path.*

3 Main section

In this section we prove that if all terminals are on integral coordinates and all length restrictions are integral, then there exists an optimal half-integral embedding. More precisely we prove that for any given feasible embedding π there exists a feasible half-integral embedding σ of at most the same cost such that the ℓ_∞ distance between the positions of a vertex in both embeddings is at most 0.5. To this end we consider a sub problem that can be formulated as a linear program based on a totally unimodular matrix.

We start with some observations on half-integral embeddings.

Proposition 2. *Every half-integral embedding has half-integral cost.*

Proof. Obviously, all edges in such an embedding have half-integral lengths and thus the total length is also half-integral. \square

Proposition 3. *In every half-integral embedding π the length of every root-terminal path has integral length.*

Proof. Let $t \in T$ and denote by P the unique root- t path in S . If P is a shortest path, then the length of P is $\|\pi(r) - \pi(t)\|_1$, which is integral. If P is not a shortest path, then by Proposition 1 there exists a component C such that moving C towards its predecessor decreases the length of P . As π is half-integral, we can move C by 0.5 towards its predecessor, reducing the length of P by 1 and obtaining a new half-integral embedding π' . Then by induction the length of P must be integral. \square

The main theorem of this section is the following.

Theorem 1. *If π is an embedding for an integral instance (S, T, r, p, l) , then there exists an half-integral embedding σ with $\max_{v \in V} \|\pi(v) - \sigma(v)\|_\infty \leq 0.5$ and $c(\sigma) \leq c(\pi)$.*

Proof. For $x \in \mathbb{R}$ we denote by $I(x)$ the smallest interval in \mathbb{R} with half-integral boundaries such that x is in the interior of the interval, that is,

$$I(x) := [\lceil 2x - 1 \rceil / 2, \lfloor 2x + 1 \rfloor / 2]. \quad (11)$$

For a point $(x, y) \in \mathbb{R}^2$ we set $I((x, y)) := I(x) \times I(y)$. We show that there exists an half-integral embedding σ with

$$\sigma(v) \in I(\pi(v)) \text{ for all } v \in V \quad (12)$$

such that $c(\sigma) \leq c(\pi)$.

Let σ be a feasible embedding for (S, T, r, p, l) of minimum cost satisfying (12). If there are several such embeddings we choose one with a minimal number of components that are not on half-integral coordinates. We denote this number by $N(\sigma)$ and prove that $N(\sigma) = 0$. Suppose that this is not the case. The idea is to move maximal components such that $N(\sigma)$ gets smaller without increasing $c(\sigma)$. As π satisfies (12), we have $c(\sigma) \leq c(\pi)$.

Let Δ_x and Δ_y be the sets of maximal x- and y-components, respectively, with respect to σ that are not on half-integral coordinates and set $\Delta := \Delta_x \dot{\cup} \Delta_y$. Then $N(\sigma) = |\Delta|$. For $C \in \Delta$ we set

$$z_C^* := \begin{cases} x(C) - \lfloor 2x(C) \rfloor / 2 & C \in \Delta_x, \\ y(C) - \lfloor 2y(C) \rfloor / 2 & C \in \Delta_y. \end{cases} \quad (13)$$

Consider a vector $z \in [0, 0.5]^\Delta$. Starting with the embedding σ and moving each component $C \in \Delta$ by $z_C - z_C^*$ we obtain a new embedding $\tau(z)$. Note that by the definition of z_C^* this embedding is half-integral if and only if $z \in \{0, 0.5\}^\Delta$. Observe that $\tau(0)$ is half-integral, but it does not necessarily satisfy the length restrictions. Since by construction $\tau(0)$ preserves the local order of σ we can apply Lemma 1 and conclude that for all $t \in T$ the length of the root- t path with respect to $\tau(0)$ is

$$d_{\tau(0)}(t) = d_\sigma(t) + \sum_{C \in \Delta: t \in R(C)} 2\text{sign}(C) \cdot (-z_C^*). \quad (14)$$

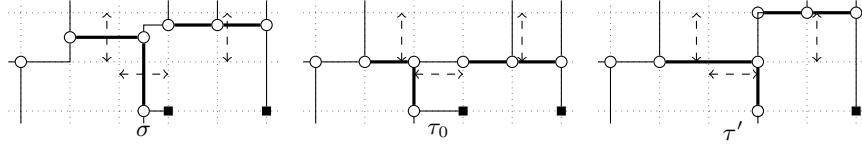


Fig. 4. Detail of an embedding σ with three maximal components not on half-integral positions. The embedding τ' preserves the local order of σ .

As $\tau(0)$ is integral, this length is also integral by Proposition 3.

Using z as a variable we can formulate a linear program reflecting the new cost of the embedding $\tau(z)$ and the length restrictions, under the assumption that $\tau(z)$ preserves the local order of σ :

$$\begin{aligned} \min & c(\sigma) + \sum_{C \in \Delta} (z_C - z_C^*) \cdot (|\Gamma_{<}^\pi(C)| - |\Gamma_{>}^\pi(C)|), \\ \text{s.t. } & d_\sigma(t) + \sum_{C \in \Delta: t \in R(C)} 2\text{sign}(C)(z_C - z_C^*) \leq l_t \quad \forall t \in T \quad (15) \\ & \text{and } 0 \leq 2z_C \leq 1 \quad \forall C \in \Delta. \quad (16) \end{aligned}$$

As $z = z^*$ is a feasible solution the linear program has an optimal solution (see also Figure 4). Substituting $2z_C$ by z'_C for all $C \in \Delta$ and using (14) we obtain the modified linear program (P'):

$$\begin{aligned} \min & \sum_{C \in \Delta} z'_C / 2 \cdot (|\Gamma_{<}^\pi(C)| - |\Gamma_{>}^\pi(C)|), \\ \text{s.t. } & \sum_{C \in \Delta: t \in R(C)} \text{sign}(C)z'_C \leq l_t - d_{\tau(0)}(t) \quad \forall t \in T \quad (17) \\ & \text{and } 0 \leq z'_C \leq 1 \quad \forall C \in \Delta. \quad (18) \end{aligned}$$

We show that the matrix A defined by the left side of the inequalities (17) is totally unimodular. Note that all entries of a column of A are either non-negative or non-positive. Thus multiplying all rows with non-positive entries by -1 we obtain a non-negative matrix where each column correspond to the characteristic vectors of $\{R(C)\}_{C \in \Delta} = \{R(C)\}_{C \in \Delta_x} \dot{\cup} \{R(C)\}_{C \in \Delta_y}$. Recall, that by Lemma 2 the sets $\{R(C)\}_{C \in \Delta_x}$ and $\{R(C)\}_{C \in \Delta_y}$ are laminar families. We conclude that the rows of A correspond to the characteristic vectors of the union of two laminar families. Edmonds [4] proved, that such matrices are totally unimodular.

Consequently, as the right hand side of (17) is integral, the constraints in (18) are integral and A is totally unimodular, there exists an optimal solution for (P') that is integral which further implies that the original LP has an half-integral optimal solution \hat{z} . But then $\tau(\hat{z})$ is also half integral and satisfies (12) and $c(\tau(\hat{z})) \leq c(\sigma)$.

If $\tau(\hat{z})$ preserves the local order of σ , then $\tau(\hat{z})$ is the embedding we are looking for. Otherwise choose $\lambda > 0$ minimal such that τ_λ defined by $\tau_\lambda(v) = \lambda\sigma(v) + (1-\lambda)\tau(\hat{z})$ preserves the local order of σ . As the cost and length functions are convex, τ_λ is a feasible embedding and $c(\tau_\lambda) \leq \lambda c(\sigma) + (1-\lambda)c(\tau(\hat{z})) \leq c(\sigma)$. Moreover, every maximal component of σ is a component of τ_λ . If $N(\tau_\lambda) = N(\sigma)$, then $\tau_{\lambda-\epsilon}$ also preserves the local order of σ for $\epsilon > 0$ small enough, contradicting the choice of λ . Thus $N(\tau_\lambda) < N(\sigma)$ contradicting the choice of σ . This finishes the proof. \square

Conclusion 2 *If all positions and length restrictions are integral, then there exists an optimal embedding that is half-integral.*

We can improve a non-optimal half-integral embedding by minor movements of vertices.

4 Dynamic programming.

A consequence of the previous section is, that any non-optimal half-integral embedding can be improved by small half-integral movements of the Steiner points.

Lemma 3. *If π is a half-integral embedding that is not optimal, then there exists a half-integral embedding π' with $\pi(v) - \pi'(v) \in \{-0.5, 0, 0.5\}^2$ for all $v \in V(S)$ and $c(\pi') \leq c(\pi) - 0.5$.*

Proof. Let σ be an optimal half-integral embedding. For $\lambda \in (0, 1)$ we define π_λ by $\pi_\lambda(v) = \lambda\pi(v) + (1-\lambda)\sigma(v)$ for all $v \in V(S)$. As π is not optimal and by the convexity of the length function, π_λ is a feasible embedding and we have $c(\pi_\lambda) \leq \lambda c(\pi) + (1-\lambda)c(\sigma) \leq c(\pi)$. Choose λ small enough such that $\max_{v \in V(S)} \|\pi(v) - \pi_\lambda(v)\|_\infty < 0.5$. Now we can apply Theorem 1 yields a half-integral embedding π' satisfying $\max_{v \in V} \|\pi(v) - \pi'(v)\|_\infty \leq \max_{v \in V} \|\pi(v) - \pi_\lambda(v)\|_\infty + \|\pi_\lambda(v) - \pi'(v)\|_\infty < 1$ and $c(\pi') \leq c(\pi_\lambda) < c(\pi)$. The claim follows by observing that π' and π are half-integral. \square

This lemma gives a direct idea for an algorithm based on dynamic programming to improve a non-optimal half-integral embedding. In the following, we interpret S as an arborescence rooted at r and denote by $I^+(v)$ the children of a vertex $v \in V(S)$. For simplicity of notation we set $\pi_\delta(v) := \pi(v) + \delta$ for $\delta \in \{-0.5, 0, 0.5\}$. Moreover, we expand the definition of length restrictions to Steiner points: Initially we set $l_t^\pi := l_t$ for all $t \in T$. For each vertex $v \in V(S)$ whose children have a length restriction, we set

$$l_v^\pi = \min_{w \in I^+(v)} l_w^\pi - \|\pi(v) - \pi(w)\|_1.$$

Given an half-integral embedding π we want to compute a half-integral embedding π' with $\pi(v) - \pi'(v) \in \{-0.5, 0, 0.5\}^2$ and $c(\pi')$ minimal. Note, that

in this case the length of every root-terminal path changes by at most $2n$. As, additionally, π' is half-integral, $l_v^{\pi'}$ is half-integral and $|l_v^{\pi'} - l_v^{\pi}| \leq 2n$ for all $v \in V(S)$.

Thus it is sufficient to compute for every vertex $v \in V(S)$, every translation $\delta \in \{-0.5, 0, 0.5\}^2$ and every possible length restriction $l \in \{l_v^{\pi} - 2n, l_v^{\pi} - 2n + 0.5, \dots, l_v^{\pi} - 2n + 2n - 0.5, l_v^{\pi} + 2n\}$ the minimum length $\gamma(v, \delta, l)$ of an embedding of the arborescence rooted at v such that v is positioned at $\pi_{\delta}(v)$ and v satisfies the length restriction l . For a terminal t we have $\gamma(t, \delta, l) = 0$ if $\delta = (0, 0)$ and $l \leq l_t$. Otherwise, we set $\gamma(t, \delta, l) = \infty$. For all other vertices $v \in V(T)$ we obviously have $\gamma(v, \delta, l) =$

$$\sum_{w \in T^+(v)} \min_{\delta' \in \{-0.5, 0, 0.5\}^2} \gamma(w, \delta', l - \|\pi_{\delta}(v) - \pi_{\delta'}(w)\|_1) + \|\pi_{\delta}(v) - \pi_{\delta'}(w)\|_1.$$

It follows, that the length of an optimal embedding π' with $\pi(v) - \pi'(v) \in \{-0.5, 0, 0.5\}^2$ is $\gamma(r, (0, 0), 0)$. This number can be computed in $O(n^2)$ time: There are $O(n^2)$ different triples (v, δ, l) for which $\gamma(v, \delta, l)$ has to be computed and each of these computations can be done in constant time.

To compute a global optimal solution, we start with the trivial embedding, where all Steiner points are positioned at the root. This solution has cost $C = \sum_{t \in T} \|p(t)\|_1$. Then we apply the dynamic programming approach as long as the cost of the newly computed embedding decrease. As the cost is reduced by at least 0.5 in every round, we must obtain an optimal embedding after $2C$ iterations. Thus our algorithm has a pseudo polynomial running time of $O(Cn^2)$. In the next section we show how to refine this approach in order to achieve a polynomial running time.

5 An optimal polynomial time algorithm

We refine the ideas of the previous sections in order to obtain a polynomial time algorithm for our problem. In the first algorithm the Steiner points are moved by at most 0.5 in each direction in every call of the dynamic programming. The idea of the refined algorithm is to move the Steiner points by 2^k for a suitable $k \in \mathbb{Z}$ in the first rounds. As soon as no improvements can be obtained by moving Steiner points by 2^k , we reduce the moving distance to 2^{k-1} and continue applying the dynamic programming. Repeating this procedure we finally move the Steiner points by 0.5, obtaining an optimal embedding. To prove the polynomial running time and to apply the results of the previous sections we have to consider slightly modified instances where all terminals are on 2^k -integral positions. Here a number $x \in \mathbb{R}$ is called 2^k -integral if $x/2^k \in \mathbb{Z}$.

First we state a trivial lemma on the existence of feasible embeddings.

Lemma 4. *There exists a feasible embedding (and thus an optimal one) for (S, T, r, p, l) if and only if $\|p(t)\|_1 \leq l_t$ for all $t \in T$.*

Proof. If there exists a feasible embedding, then obviously $\|p(v)\|_1 \leq l(v)$ for all $v \in T$. If on the other hand $\|p(v)\|_1 \leq l(v)$, then placing all internal vertices on

the position of the root is a feasible embedding satisfying the length restrictions. \square

For each $k \in \mathbb{N}$ we define a new instance $I_k := (S, T, r, p_k, l_k)$ on the same set of terminals and the same topology, but with new positions

$$p_k(v) = (2^k \lfloor p_x(v)/2^k \rfloor, 2^k \lfloor p_y(v)/2^k \rfloor)$$

for all $v \in T$ and length restrictions

$$l_k(v) = 2^k \lfloor (l(v) - \|p(v) - p_k(v)\|_1)/2^k \rfloor$$

for $v \in T$. In other words we move each terminal towards the root onto the next 2^k -integral position and round each length restriction to the next lower multiple of 2^k .

If there exists a feasible embedding for (S, T, r, p, l) , then there exists also a feasible embedding for (S, T, r, p_k, l_k) : To show this it is sufficient to prove that $\|p_k(v)\|_1 \leq l_k(v)$ for all $v \in T$ by Lemma 4. By the choice of p_k and l_k we have

$$\|p_k(v)\|_1 = \|p(v)\|_1 - \|p(v) - p_k(v)\|_1 \leq l(v) - \|p(v) - p_k(v)\|_1 \leq l_k(v). \quad (19)$$

Set $m := \min\{m \in \mathbb{N} : |p_x(v)| < 2^m \text{ and } |p_y(v)| < 2^m \forall v \in V(S)\}$. Thus m is the smallest $m \in \mathbb{N}$ such that $p_m(v) = (0, 0)$ for all $v \in V(S) \setminus T$.

Remark 1. The number m is polynomially bounded in the size of the instance.

In (S, T, r, p_m, l_m) all terminals are placed at the position of the root. Thus placing all internal vertices to that position yields a trivial optimal solution of length 0.

Now we compute by induction an optimal embedding for $(S, T, r, p_{k-1}, l_{k-1})$ given an optimal embedding for (S, T, r, p_k, l_k) .

As m is polynomially bounded in the size of the input, each iteration can be computed in polynomial time and $(S, T, r, p, l) = (S, T, r, p_0, l_0)$, we obtain a optimal solution in polynomial time.

Lemma 5. *Denote by σ_k an optimal solution for I_k for all $k \in \mathbb{N}$. Then for $k \in \mathbb{N}$ we have $c(\sigma_{k+1}) \leq c(\sigma_k) + 6n2^k$.*

Proof. Starting with σ_k we construct a feasible embedding for I_{k+1} . By Lemma 1 we can assume w.l.o.g. that all internal vertices of S are on 2^{k-1} -integral positions in σ_k . We define π by setting $\pi(t) = p_{k+1}(t)$ for $t \in T$ and $\pi(v) = \sigma_k(v)$ for $v \in V(S) \setminus T$. By this setting we have

$$\|\pi(t) - \sigma_k(t)\|_1 = \|p_{k+1}(t) - p_k(t)\|_1 \leq 2^{k+1}$$

for all $t \in T$. Thus

$$c(\pi) \leq c(\sigma_k) + n2^{k+1} \quad (20)$$

and the length of each root-terminal path increased by at most 2^{k+1} . As $l_k(t) \leq l_{k+1}(t) + 2^{k+1}$ for all $t \in T$, we conclude that for each $t \in T$ the length restriction $l_{k+1}(t)$ is hurt in π by at most 2^{k+2} :

$$\sum_{e \in E[r, t]} \pi(e) \leq \sum_{e \in E[r, t]} \sigma_k(e) + 2^{k+1} \leq l_k(t) + 2^{k+1} \leq l_{k+1}(t) + 2^{k+2}. \quad (21)$$

Now we move components towards their predecessors, until all length restrictions are satisfied. To this end, denote by Δ the set of all maximal components, that do not contain the root r . Moving all components $C \in \Delta$ by 2^{k-1} towards its predecessors we obtain a new feasible embedding π' with $c(\pi') \leq c(\pi) + n2^k$. Moreover, the length of every root-terminal path that has not been a shortest one with respect to π is reduced by at least 2^k . Repeating this process with the new embedding at most 3 times yields a feasible embedding π^* for I_{k+1} . We conclude $c(\pi^*) \leq c(\pi) + 4n2^k$. Together with (20) we conclude $c(\pi^*) \leq c(\sigma_k) + 6n2^k$. We finish the proof by observing, that as π^* is feasible for I_{k+1} , the embedding σ_{k+1} cannot be longer. \square

Combining the observation of the previous Lemma we obtain our main result.

Theorem 3. *The rectilinear Steiner tree embedding problem with length restrictions can be solved in polynomial time by a combinatorial algorithm.*

Proof. Let $I = (S, T, r, p, l)$ be an instance of the problem. First we calculate m as above. m is polynomially bounded in the size of the input. Now we have a polynomial number of instances I_k , $k \in \{1, \dots, m\}$. For I_m we have the trivial embedding σ_m with $\pi_m(v) = 0$ for all $v \in V$.

Let σ_{k+1} be an optimal embedding for I_{k+1} . Then π defined as $\pi(v) = p_k(v)$ for $v \in T$ and $\pi(v) = \sigma_{k+1}(v)$ otherwise is a feasible embedding for I_k . By Lemma 5, $c(\pi) \leq c(\sigma_{k+1}) + n2^k \leq O_k + 7n2^k$ where O_k denotes the optimal length of an embedding for I_k . Moreover, I_k is an 2^k integral instance. Thus applying the dynamic programming from the previous section at most $14n$ times we obtain an optimal solution σ_k for I_k . We conclude that computing σ_k from σ_{k+1} requires at most time $O(n^3)$. By induction we get an optimal solution σ for $I_1 = (S, T, r, c, l)$. The total running time is $O(mn^3)$ where $m = \lceil \max\{|p_x(t)|, |p_y(t)| : t \in T\} + 1 \rceil$. \square

Obviously, every feasible solution for I_k corresponds to a feasible solution for I . Moreover, all Steiner points of such embeddings are one 2^{k-1} -integral positions. Due to this observation, the implementation of the algorithm can be modified in order to decrease the number of dynamic programming steps in practice. Instead of computing an optimal solution for I_k , we are looking for an embeddings of minimal cost for the original instance I , but all Steiner points have to be on 2^{k-1} -integral coordinates. We use the dynamic programming steps as described in order to improve a given embeddings, but now the cost of each solution is computed using the original positions of the terminals and consider

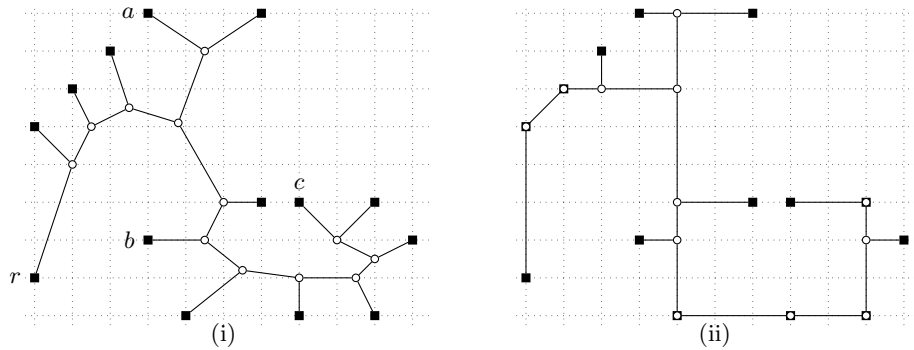


Fig. 5. Instance for the Steiner tree embedding problem (i) and an optimal embedding if there are no length restrictions (ii).

the original length restrictions. Using this method, the number of dynamic programming steps performed in the algorithm get very small. It turns out, that in practice, the number of dynamic programming calls is constant for each k in the most cases.

Figures 5 and 6 show how the algorithm works on an example. Figure 5 (i) shows the instance and Figure 5 (ii) an optimal embedding of length 35 if there are no length restrictions. In Figure 6 the embeddings computed by our algorithm are shown. As input we used the instance from Figure 5 with length restrictions $l_a = 10$, $l_b = 11$ and $l_c = 20$. As $\max\{|\pi_x(t)|, |\pi_y(t)| : t \in T\} = 10$ we have $m = 5$. Thus the algorithm begins with an embedding where all Steiner points are 2^{m-1} -integral (Figure (i)). Proceeding with $k = 4$ (Figure (ii)) to $k = 0$ (Figure (vi)). The last one is the final optimal embedding of length of 37.5. For each k the dynamic programming is called at most twice, the first time the length is reduced, the second time an embedding of the same cost is computed, proving, that it is an optimal one.

References

1. Christoph Bartoschek, Stephan Held, Jens Maßberg, Dieter Rautenbach, and Jens Vygen. The repeater tree construction problem. *Information Processing Letters*, 110(24):1079–1083, 2010.
2. Ulrich Brenner and Jens Vygen. Analytical methods in VLSI placement. *Handbook of Algorithms for VLSI Physical Design Automation (C.J. Alpert, D.P. Mehta, S.S. Sapatnekar, eds.)*, Taylor and Francis, 2008.
3. A. Victor Cabot, Richard L. Francis, and Michael A. Stary. A network flow solution to a rectilinear distance facility location problem. *AIIE Transactions*, 2(2):132–141, 1970.
4. Jack Edmonds. Submodular functions, matroids and certain polyhedra. In Gordon and Breach, editors, *Combinatorial Structures and Their Applications*, pages 68–87, New York, 1970.

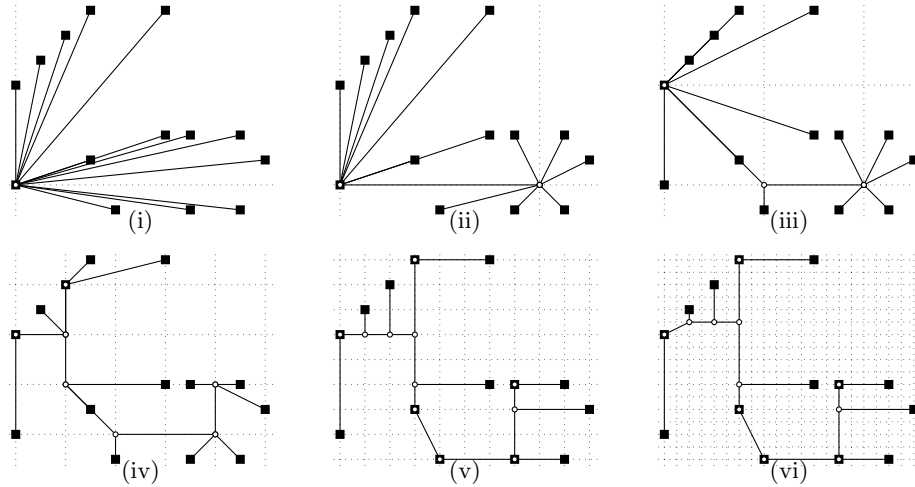


Fig. 6. Run of the algorithm on the instance shown in Figure 5 (i) with length restrictions $l_a = 10$, $l_b = 11$ and $l_c = 20$. Figure (vi) shows the final optimal solution.

5. Michael R. Garey and David S. Johnson. The rectilinear Steiner tree problem is NP-complete. *SIAM Journal on Applied Mathematics*, 32(4):826–834, 1977.
6. E.N. Gilbert and H.O. Pollak. Steiner minimal trees. *SIAM Journal on Applied Mathematics*, 16(1):1–29, 1968.
7. Maurice Hanan. On Steiner’s problem with rectilinear distance. *SIAM Journal on Applied Mathematics*, 14(2):255–265, 1966.
8. Frank K. Hwang. A linear time algorithm for full Steiner trees. *Operations Research Letters*, 4(5):235–237, 1986.
9. Tao Jiang and Lusheng Wang. Computing shortest networks with fixed topologies. In *Advances in Steiner Trees*, volume 6 of *Combinatorial Optimization*, pages 39–62. Springer US, 2000.
10. Alexander V. Karzanov. Minimum cost multiflows in undirected networks. *Mathematical programming*, 66(1-3):313–325, 1994.
11. Sailesh K. Rao, P. Sadayappan, Frank K. Hwang, and Peter W. Shor. The rectilinear Steiner arborescence problem. *Algorithmica*, 7(1-6):277–288, 1992.
12. Weiping Shi and Chen Su. The rectilinear Steiner arborescence problem is NP-complete. *SIAM Journal on Computation*, 35(3):729–740, 2005.
13. Guoliang Xue and Krishnaiyan Thulasiraman. Computing the shortest network under a fixed topology. *Computers, IEEE Transactions on*, 51(9):1117–1120, 2002.
14. Martin Zachariasen. Comment on “Computing the shortest network under a fixed topology”. *IEEE Transactions on Computers*, 55(6):783–784, 2006.